

SC2006: Software Engineering

Lab 1 Deliverables

DateGoWhere

Team SPAZM

Lu Zihao (U2122553H)

Markus Lim En Cheng (U2222471K)

Paviru Vidwan Perera (U2222381L)

Tam Yik Lock (U2222001C)

Tan Shao Chong (U2222070J)

Table of Contents

<i>Introduction</i>	<i>3</i>
<i>Functional Requirements</i>	<i>4</i>
<i>Non-Functional Requirements.....</i>	<i>7</i>
Performance	7
Safety	7
Reliability.....	7
Security	7
Usability	7
Supportability	7
<i>Data Dictionary</i>	<i>8</i>
<i>Use Case Model.....</i>	<i>9</i>
Use Case Diagram	9
Use Case Descriptions	10
<i>UI Mock-ups</i>	<i>19</i>

Introduction

DateGoWhere is a web application that facilitates the planning of dates with significant others, families, and friends. Users can share their date itineraries on the app's catalogue for others to view and use. Users can also rate other date itineraries in the catalogue, as well as leave their reviews.

Functional Requirements

1. Users shall be able to query the system.

- 1.1 Users shall be able to query the system by using their live location or destination location name.
- 1.2 The application must check if the User has given GPS permission.
 - 1.2.1 If GPS permission is not given, the User must be prompted to give GPS.
 - 1.2.2 Should the request for the GPS permission be denied, the application should not prompt the user again.
- 1.3 The application should allow the user to search the desired destination locations.
 - 1.3.1 The user must input their query in text format, and it can include alphanumeric characters.
- 1.4 A response for a valid query location for both live location and location name will return all business locations in database in a 3km radius of the user using Google Maps API
 - 1.4.1 The system will also show the location on the map, denoted with a red pin.
 - 1.4.2 The system will re-prompt the user for a valid location should the query be invalid.
- 1.5 All responses must have the same format.
 - 1.5.1 All responses must contain user data, location data followed by business data.
 - 1.5.1.1 The user data for a valid location must contain the latitude and longitude of the user.
 - 1.5.1.2 The location data for a valid location must contain the latitude and longitude of the business and the address of the business.
 - 1.5.1.3 The business data for a valid business must contain the business type, business name, a picture of the storefront, business contact information and operational hours of the business.
 - 1.5.1.4 If any data is unavailable, the system shall replace it with None.
- 1.6 Users shall be able to view from the list of businesses in the system's response.
 - 1.6.1 Users shall be able to select desired locations to visit.
 - 1.6.1.1 If selected, the system will provide a map button that redirects the user to the Google Maps application, which provides detailed directions to the location.

2. Users shall be able to authenticate their account.

- 2.1 Users must input their email address into an input textbox to create an account.
 - 2.1.1 The system must check that the email address that the user input is valid.
 - 2.1.1.1 If the email address is not valid, the system must alert the user and prompt an "Invalid email" message.
 - 2.1.2 Repeat uses of an email address will not be allowed.
 - 2.1.2.1 The system shall check with the database for repeat uses and the system must alert the user and prompt an "Email is already in use" message.
 - 2.1.3 Users must input a 12-24 character long alphanumeric password into a textbox.

- 2.1.3.1 The system shall check if the password contains at least 1 uppercase character and 1 special character.
 - 2.1.3.2 The system shall alert the user if the password does not fulfill the above criteria, prompting a "Invalid Password" message.
 - 2.2 Users shall be able to log into their account using either the username or email address accompanied by the valid password.
 - 2.2.1 The system must be able to display a "Forgot Password" field.
 - 2.2.1.1 If the user clicks on the field, the system must prompt the email address and username of the user.
 - 2.2.1.2 If the email address matches the username, the system shall provide a link to their email to reset their password.
 - 2.3 Users shall be able to opt in for 2 factor authentication.
 - 2.4 After users created their account, the user shall be redirected to our landing page.
- 3. Authenticated users shall be able to share a list of curated date ideas.
 - 3.1 Curated date ideas should be a set of locations/activities.
 - 3.1.1 The list of locations must contain a minimum of 1 location and a maximum of 5.
 - 3.1.2 Each location in the list must have the same format.
 - 3.1.2.1 Each location must contain a picture of the location uploaded by the user, followed by the name of the location followed by the exact location.
 - 3.1.2.2 Each location can contain an additional brief description of the place, with a maximum of 500 words.
 - 3.2 The system shall store the locations in the same format.
 - 3.2.1 The system shall store the location data followed by business/activity data.
 - 3.2.1.1 The location data for a valid location must contain the latitude and longitude of the business and the address of the business.
 - 3.2.1.2 The business data for a valid location must contain the business type, business name, a picture of the storefront, business contact and operational hours of the business.
 - 3.3 The system shall be able to sort curated date ideas following the number of likes that each post has, either in increasing or decreasing order
 - 3.4 The system shall be able to sort curated date ideas according to the price range
 - 3.4.1 The User shall be able to indicate a price range through a drop down box, indicating both minimum price and maximum price
 - 3.4.2 The system shall be able to sort through all curated ideas and display all ideas starting with ideas that most closely match the minimum price selected by the user.
 - 3.4.3 If there are no items that fall in the selected price range, the system shall display "No items found".
 - 3.5 Authenticated users can report curated date ideas for inaccurate information.
 - 3.5.1 The system shall keep track of the number of reports of each post and flag it for review should the number of reports exceed 7% of the MAU.

- 4. Authenticated users shall be able to review the list of curated date ideas.
 - 4.1 Users shall be able to leave comments on the date ideas for other users to view.
 - 4.1.1 The system shall store comments in the same format, with user data first followed by comment data.
 - 4.1.1.1 User data shall include the username of the commenting user.
 - 4.1.1.2 Comment data shall be stored only as plaintext, not exceeding 100 words.
 - 4.1.1.3 Each user shall be limited to 2 comments per date idea.
 - 4.1.2 Users shall be able to report comments for offensive language, irrelevance, or inaccurate information.
 - 4.2 Users shall be able to like posts and provide a rating from 1 to 10.

Non-Functional Requirements

Performance

- The site should be able to load within 5 seconds.
- The search function should return results within 5 seconds.

Safety

- Highly reported (> 7% of MAU) posts shall be brought up for review and deleted should it contain discriminatory, obscene or harmful content that does not abide to our community guidelines.
- The system will check comments when they are posted against a database of banned phrases and words, preventing comments containing discriminatory, obscene, or harmful content that does not abide to community guidelines.

Reliability

- Critical bugs must be addressed within 48 hours of reporting.
- After a system reboot, the full system functionality must be restored within 5 minutes.
- There shall be no more than 1 hour of planned downtime per year.

Security

- Sensitive user data stored in the database must be encrypted using AES-256 or equivalent encryption algorithms.
- Users will automatically be logged out after 10 minutes of inactivity (session timeout).

Usability

- Help messages must be displayed in the local language according to the user's locale.

Supportability

- The code will be organized into Modular segments, with a well-defined structure and camel-cased naming conventions with relevant comments, to allow for easy readability and maintenance.
- The software shall make use of a dependency management system to manage and organise any third-party libraries and components used. This allows for easy updates and modification of software.

Data Dictionary

Term	Definition
Date	A planned and intentional social outing or activity that allows individuals to spend quality time together and strengthen their relationships.
Itinerary	A pre-planned and organised schedule or list of activities that outlines the events, locations, and timing for the date.

Use Case Model

Use Case Diagram



Use Case Descriptions

Use Case ID:	UCX - 001		
Use Case Name:	Upload Date Idea		
Created By:	Paviru	Last Updated By:	Paviru
Date Created:	03/09/23	Date Last Updated:	03/09/23

Actor:	User
Description:	User can upload their date ideas onto the app for others to view. They can include information such as location, time required, price, and activities to be done. With more uploaded ideas, other users will be able to gain more inspiration for dates.
Preconditions:	User is logged in and authenticated.
Postconditions:	Date idea is successfully uploaded to the app.
Priority:	High
Frequency of Use:	Frequent
Flow of Events:	<ol style="list-style-type: none"> 1. User selects the "Upload Date Idea" option. 2. User submits the date idea. 3. The app processes the information provided. 4. If uploading is successful, the date idea is added to the app's database. 5. The app notifies the user of successful upload.
Alternative Flows:	AF-S6. If uploading fails (e.g., missing information), the app displays an error message, and the user is prompted to correct the details. App returns to Step 2.
Exceptions:	-
Includes:	<ol style="list-style-type: none"> 1. Authentication required prior to uploading date idea.
Special Requirements:	-
Assumptions:	-
Notes and Issues:	-

Use Case ID:	UCX - 002		
Use Case Name:	Delete/Edit Date Idea		
Created By:	Paviru	Last Updated By:	Paviru
Date Created:	03/09/23	Date Last Updated:	03/09/23

Actor:	User
Description:	User can edit or delete their uploaded date ideas. Date ideas specific to events may be outdated or unavailable, hence ideas must be updated to remain relevant.
Preconditions:	User is logged in and authenticated.
Postconditions:	Date idea is successfully deleted or edited.
Priority:	Low
Frequency of Use:	Seldom
Flow of Events:	<ol style="list-style-type: none"> 1. User selects a date idea they have previously uploaded. 2. User chooses to either edit or delete the date idea. 3. If editing, the user makes changes to the date idea details and submits the updated information. 4. If deleting, the user confirms to delete the idea. 5. The app processes the edit or delete request. 6. The app notifies the user of the success or failure of the request.
Alternative Flows:	AF-S5. If processing fails (e.g., missing information), the app displays an error message, and the user is prompted to correct the details. App returns to Step 3 or 4.
Exceptions:	-
Includes:	<ol style="list-style-type: none"> 1. Authentication required prior to modifying date idea.
Special Requirements:	-
Assumptions:	-
Notes and Issues:	-

Use Case ID:	UCX - 003		
Use Case Name:	Browse Date Idea		
Created By:	Paviru	Last Updated By:	Paviru
Date Created:	03/09/23	Date Last Updated:	03/09/23

Actor:	User
Description:	User can browse various date ideas uploaded from other Users. Users can filter date ideas by theme, location and price. Users can click date idea to view the location, price, activities done during the date and more information.
Preconditions:	User has app downloaded.
Postconditions:	User views a list of date ideas based on applied filters and views further information on interested date idea.
Priority:	High
Frequency of Use:	Frequent
Flow of Events:	<ol style="list-style-type: none"> 1. User selects the "Browse Date Ideas" option. 2. User applies filters (e.g., price, distance, theme) to narrow down the list of date ideas. 3. The app retrieves and displays date ideas based on the applied filters. 4. User can click on a date idea to view more information. 5. User views a map feature with location of date idea and a list of information such as price and activities to be done.
Alternative Flows:	AF-S3. If no date ideas match the applied filters, the app informs the user accordingly. App returns to Step 2.
Exceptions:	-
Includes:	<ol style="list-style-type: none"> 1. Get shared date ideas from other users.
Special Requirements:	-
Assumptions:	-
Notes and Issues:	-

Use Case ID:	UCX - 004		
Use Case Name:	Plan Date Idea		
Created By:	Paviru	Last Updated By:	Paviru
Date Created:	03/09/23	Date Last Updated:	03/09/23

Actor:	User
Description:	User can select interested date idea and input their home address, partner's address, and other locations he/she wants to visit. The app will then suggest the fastest route based on Google Maps route API.
Preconditions:	User has app downloaded.
Postconditions:	User receives a planned route with estimated time required.
Priority:	Medium
Frequency of Use:	Frequent
Flow of Events:	<ol style="list-style-type: none"> 1. User selects a date idea they are interested in. 2. User chooses the "Plan Date Idea" option. 3. User inputs the locations they plan to visit during the date, in sequence. 4. The app uses the Google Maps API to calculate and display an end-to-end route from the user's location to the date idea's location and other included locations. 5. The app estimates the time required for the planned date. 6. User can choose to discard, save or share the planned route.
Alternative Flows:	AF-S4. If the Google Maps API is unable to provide location for the User input location, an error message will be displayed. App returns to Step 3.
Exceptions:	-
Includes:	<ol style="list-style-type: none"> 1. Google Maps Routes API to plan routes. 2. Google Maps Places API to get valid businesses 3. Share Date Idea is extended to provide option for User to share route with their partner.
Special Requirements:	-
Assumptions:	-
Notes and Issues:	Time estimation will be based on expected duration at the intended time of date. Exact time will be unable as User will likely be planning date beforehand.

Use Case ID:	UCX – 005		
Use Case Name:	Review Date Idea		
Created By:	Paviru	Last Updated By:	Paviru
Date Created:	03/09/23	Date Last Updated:	03/09/23

Actor:	User
Description:	User can review a date idea after he/she has gone through it. The review will be done in 2 steps. User can provide a star rating out of 10. User can also optionally type out a review paragraph of maximum
Preconditions:	User is logged in and authenticated.
Postconditions:	User submits a review for the date idea.
Priority:	Medium
Frequency of Use:	Seldom
Flow of Events:	<ol style="list-style-type: none"> 1. User selects a date idea they have completed. 2. User chooses the "Review Date Idea" option. 3. User provides a rating out of 10 stars. 4. User can choose to include a written review of the date idea. 5. If User chooses to include a written review, he/she will be prompted to type his review. 6. The app processes the review. 7. The app notifies the user that the review has been successfully submitted.
Alternative Flows:	AF-S6. If review cannot be processed (e.g. Too many characters for review, the app will display an error message. App returns to Step 5.
Exceptions:	-
Includes:	<ol style="list-style-type: none"> 1. Authentication required prior to reviewing date idea.
Special Requirements:	-
Assumptions:	User has completed the date idea.
Notes and Issues:	-

Use Case ID:	UCX - 006		
Use Case Name:	Share Date Idea		
Created By:	Paviru	Last Updated By:	Paviru
Date Created:	03/09/23	Date Last Updated:	03/09/23

Actor:	Date Seeker
Description:	User can share either the date idea or the route planned with partner or friends. The sharing will be done through social media platforms such as Telegram, WhatsApp, or Email.
Preconditions:	User has app downloaded.
Postconditions:	User shares the date idea to their intended recipient.
Priority:	Medium
Frequency of Use:	Frequent
Flow of Events:	<ol style="list-style-type: none"> 1. User selects a date idea or the date route planned they want to share. 2. User chooses the "Share Date Idea" option. 3. The app provides options for sharing via various social media platforms. 4. User selects a platform and shares the date idea with their partner or friends. 5. The app processes the sharing. 6. The app notifies the user that the idea has been shared successfully.
Alternative Flows:	AF-S5. If there are issues with sharing on social media (e.g., authentication failure), the app informs the user. App returns to Step 4.
Exceptions:	-
Includes:	-
Special Requirements:	-
Assumptions:	-
Notes and Issues:	-

Use Case ID:	UCX - 007		
Use Case Name:	Authenticate User		
Created By:	Paviru	Last Updated By:	Paviru
Date Created:	03/09/23	Date Last Updated:	03/09/23

Actor:	Date Seekers and Date Suggesters
Description:	All Users must log in to their account before they can upload ideas or review ideas. This is to provide legitimacy and accountability of ideas uploaded and the quality of reviews.
Preconditions:	User is not authenticated.
Postconditions:	User successfully logs in or registers an account.
Priority:	High
Frequency of Use:	Frequent
Flow of Events:	<ol style="list-style-type: none"> 1. User selects the "Log In" or "Register" option. 2. If User selects "Register", App requests for necessary information such as email and password for registration. 3. If User selects "Login", App requests for login credentials for existing users. 4. The app validates the information provided. 5. If validation is successful for registration, a new account is created. 6. If validation is successful for login, the user gains access to their account. 7. The app notifies the user of successful authentication or registration.
Alternative Flows:	<p>AF-S5 - If validation fails (e.g., invalid email), the app displays an error message and prompts the user to correct the details. App returns to Step 2.</p> <p>AF-S6</p> <p>- If validation fails due to incorrect username/email, the app displays an error message and prompts the user to register instead. App returns to Step 2.</p> <p>- If validation fails due to incorrect password, the app displays an error message and prompts the user to re-enter password. App returns to Step 3.</p>
Exceptions:	EX-S6 – If User inputs password incorrectly more than 5 times, the app displays message "Account suspended". A reset password email is sent to the registered email, for the User to login again.
Includes:	-
Special Requirements:	-
Assumptions:	-
Notes and Issues:	-

Use Case ID:	UCG - 001		
Use Case Name:	Get Valid Businesses		
Created By:	Paviru	Last Updated By:	Paviru
Date Created:	03/09/23	Date Last Updated:	03/09/23

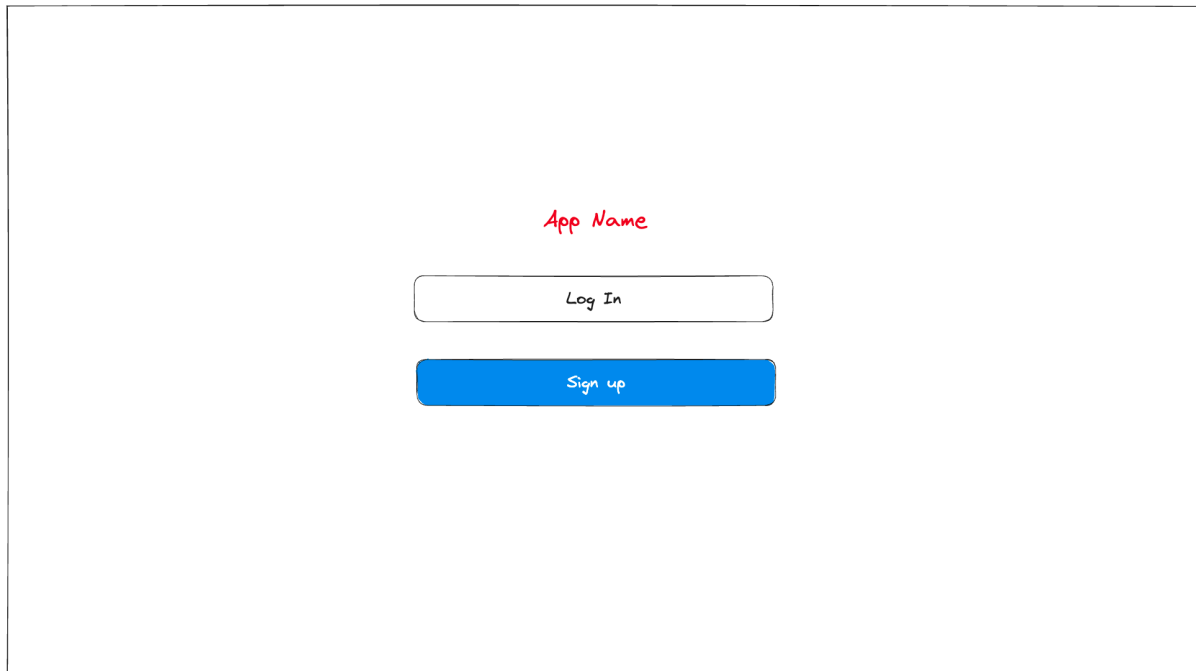
Actor:	Google Maps
Description:	Retrieve location information for an input address on the integrated Map feature. This will help users to plan their date more easily.
Preconditions:	User has input an address/postal code/place name into the App.
Postconditions:	User is successfully displayed input location on the Map.
Priority:	High
Frequency of Use:	Frequent
Flow of Events:	<ol style="list-style-type: none"> 1. User inputs the location either through text input or allowing access to location data. 2. App provides input to Google Maps Places API. 3. If Google Maps Location API can find location, it provides details of location to App. 4. App displays location as a marker on the Integrated Map.
Alternative Flows:	AF-S3 - If Google Maps Places API cannot find location (e.g., invalid postal code), the app displays an error message and prompts the user to enter another location. App returns to Step 2.
Exceptions:	-
Includes:	-
Special Requirements:	Google Maps Places API
Assumptions:	-
Notes and Issues:	-

Use Case ID:	UCG - 002		
Use Case Name:	Get Routes		
Created By:	Paviru	Last Updated By:	Paviru
Date Created:	03/09/23	Date Last Updated:	03/09/23

Actor:	Google Maps
Description:	Retrieve optimal route for an input address and location listed on date idea on the integrated Map feature. This will help users to plan their date more easily.
Preconditions:	User has input at least 2 address/postal code/place name into the App.
Postconditions:	User is successfully displayed route going through all input locations on the Map.
Priority:	High
Frequency of Use:	Frequent
Flow of Events:	<ol style="list-style-type: none"> 1. User inputs multiple locations either through text input or allowing access to location data. User must put the locations in sequence of intended visit. 2. App provides input to Google Maps Routes API. 3. If Google Maps Routes API can find all locations and valid routes to all locations, it provides details of route to App. 4. App displays route on the Integrated Map.
Alternative Flows:	<p>AF-S3</p> <ul style="list-style-type: none"> - If Google Maps Routes API cannot find location (e.g., invalid postal code), the app displays an error message and prompts the user to enter another location. App returns to Step 2. - If Google Maps Routes API cannot find a valid route (e.g., inaccessible location), the app displays an error message and prompts the user to enter another location. App returns to Step 2.
Exceptions:	-
Includes:	-
Special Requirements:	Google Maps Location API
Assumptions:	-
Notes and Issues:	-

UI Mock-ups

Onboarding



A mockup of an onboarding screen. It features a red text label "App Name" at the top center. Below it are two buttons: a white button with a black border labeled "Log In" and a solid blue button labeled "Sign up".

App Name

Log In

Sign up

Landing



A mockup of a landing screen. In the top right corner is a user profile icon. On the left side, there are three buttons stacked vertically: "Create a new date plan", "View/edit my date plans", and "Browse date ideas". On the right side, the text "Welcome, UserName" is displayed.

Create a new date plan

View/edit my date plans

Browse date ideas

Welcome, UserName

Create Plan

Your new date plan name:

Your first date activity:

Start

End

Activity

Location

Description

Budget

Discard

Add activity

Save draft

Submit

Creating your new date plan

Your new date plan name:

Romantic Night By The Bay

Your first date activity:

1600

1800

Picnic

 Choose on map

Description

Budget

Discard

Save activity

Add another activity

Save draft

Submit

Browse Plans

Q Search Date Ideas

Romantic Night By The Bay

A romantic night for couples with beautiful views of the bay area.

\$50-\$60 6 hours 10 Likes

#Picnic #Satay #Nightview

Romantic Night By The Bay

A romantic night for couples with beautiful views of the bay area.

\$50-\$60 6 hours 10 Likes

#Picnic #Satay #Nightview

Romantic Night By The Bay

A romantic night for couples with beautiful views of the bay area.

\$50-\$60 6 hours 10 Likes

#Picnic #Satay #Nightview

Romantic Night By The Bay

A romantic night for couples with beautiful views of the bay area.

\$50-\$60 6 hours 10 Likes

#Picnic #Satay #Nightview

Romantic Night By The Bay

A romantic night for couples with beautiful views of the bay area.

\$50-\$60 6 hours 10 Likes

#Picnic #Satay #Nightview

Romantic Night By The Bay

A romantic night for couples with beautiful views of the bay area.

\$50-\$60 6 hours 10 Likes

#Picnic #Satay #Nightview

Filter Plans

Q Search Date Ideas

Filters

Price range

S\$ — S\$

Location

Choose on map

Custom tags

View Plan

Romantic Night By The Bay

A romantic night for couples with beautiful views of the bay area.

16:00-18:00 Picnic @ Marina Barrage

Author description

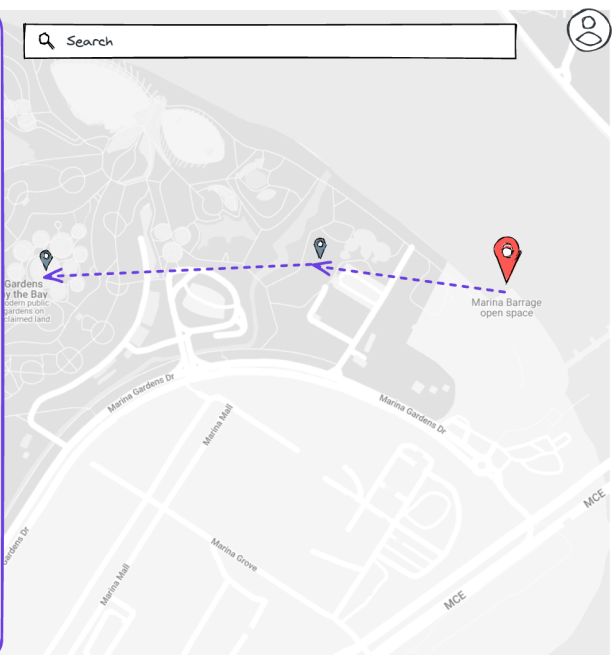

Estimated Damage: \$20

18:15-20:00 Dinner @ Satay By The Bay

20:10-22:00 Hangout @ Gardens By The Bay

[See community reviews](#)

[Review this date](#)



Review Plan

Romantic Night By The Bay

A romantic night for couples with beautiful views of the bay area.

16:00-18:00 Picnic @ Marina Barrage

Author description

Estimated Damage: \$20

18:15-20:00 Dinner @ Satay By The Bay

20:10-22:00 Hangout @ Gardens By The Bay

[See community reviews](#)

[Review this date](#)



How would you rate this date?

Submit